



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,923	10/23/2003	Brendan McKeon	3382-66149	9047
26119 7590 05/17/2007 KLARQUIST SPARKMAN LLP 121 S.W. SALMON STREET SUITE 1600 PORTLAND, OR 97204			EXAMINER SHIH, HAOSHIAN	
			ART UNIT 2173	PAPER NUMBER
			MAIL DATE 05/17/2007	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

Application No.

10/692,923

Applicant(s)

MCKEON ET AL.

Examiner

Haoshian Shih

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 25 April 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-35 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-9 and 21-35 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☒ Claim(s) 10-20 are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 23 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. Claims 1-9 and 21-35 are pending in this application and have been examined in response to application filed on 10/23/2003.

### **RESPONSE TO APPLICANT'S ELECTION**

The Examiner has divided the claims into three groups:

- I. Claims 1-9 are drawn to "Generating database or data structure", classified in class 707, subclass 102.
- II. Claims 10-20 are drawn to "Query processing", classified in class 707, subclass 3.
- III. Claims 21-35 are drawn to "Database schema or data structure", classified in class 707, subclass 100.

Applicant elects group III and proposes revised claim groups that are believe to more appropriately group the claims according to their status in the art. Applicant combines groups I and III.

The Examiner agrees with the grouping of groups I and III.

### ***Specification***

2. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. **It is important that the abstract not exceed 150 words in length** since the space provided for the abstract on the computer tape used by the printer is limited.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. **Claims 1-5, 7-9, 21-23, 27-30 and 34-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Der Linden et al. (Van, US 2005/0050011 A1) in view of Chow et al. (Chow, US 5,642,511**

5. As to **INDEPENDENT** claim 1, Van discloses a method of generating identifier data for persistently identifying an element of interest of a source computer program, the method comprising:

receiving data indicative of the element of interest from a first software component (pg.2, [0019], elements of interest are received as a query); and

in response to receiving the data indicative of the element of interest, generating an element path identifier of the element of interest for persistently identifying the element of interest and returning at least the element path identifier to the first software component (pg.5, [0056], "node name", "namespace").

Van does not disclose the element of interest is a graphical user interface element.

In the same field of endeavor, Chow discloses representing a plurality of user interface elements in a tree structure.

It would have been obvious to one of ordinary skill in the art, having the teaching of Van and Chow before him at the time the invention was made, to modify the identifying and communicating of elements taught by Van to include representing of user interface elements taught by Chow with the motivation being to provide information to a requesting component more efficiently (pg.1, [0007], Van).

6. As to **INDEPENDENT** claim 21, Van discloses, at least one computer-readable medium having stored thereon computer-executable instructions related to a function responsive to a function call from a first software component (pg.1, [0006]; a client application query system is presented), the function comprising:

an input parameter representing an element of interest in a source computer program (pg.2, [0019], elements of interest are received as a query) ;

an output parameter representing an element path identifier for persistent identification of the element of interest (pg.5, [0056], "node name", "namespace").

executable software for receiving the input parameter representing a user interface element of interest and in response, generating the output parameter representing an element path identifier of the element of interest (pg.2, [0019]; the software executes Xpath or Xquery expression to output the element path identifier).

Van does not disclose the element of interest is a graphical user interface element.

In the same field of endeavor, Chow discloses representing a plurality of user interface elements in a tree structure.

It would have been obvious to one of ordinary skill in the art, having the teaching of Van and Chow before him at the time the invention was made, to modify the identifying and communicating of elements taught by Van to include representing of user interface elements taught by Chow with the motivation being to provide information to a requesting component more efficiently (pg.1, [0007], Van).

7. As to **INDEPENDENT** claim 28, claim 28 incorporates substantially similar subject matter as claimed in 21, and is rejected along the same rationale.

8. As to **INDEPENDENT** claim 35. Van discloses a system for generating element path identifiers of elements and later searching for the elements using the element path identifiers, the system comprising:

An API module comprising a first set of APIs related to passing function calls for generating the element path identifiers (pg.2, [0023], last 3 lines; [0024]; “node tree generator”)

and a second set of APIs related to passing function calls for searching for the elements of the graphical user interface using the element path identifiers (pg.2, [0019], lines 8-12; the search is processed via the execution of an Xpath or Xquery expression); and

an element path engine responsive to the function calls for generating the element path identifiers (pg.2, [0023], last 3 lines; [0024]; “node tree generator”) and to the function calls for searching for the elements of the graphical user interface using the element path identifiers (pg.2, [0019], lines 8-12; the search is processed via the execution of an Xpath or Xquery expression). Van does not disclose the elements are graphical user interface elements.

In the same field of endeavor, Chow discloses representing a plurality of user interface elements in a tree structure.

It would have been obvious to one of ordinary skill in the art, having the teaching of Van and Chow before him at the time the invention was made, to modify the identifying and communicating of elements taught by Van to include representing of user interface elements taught by Chow with the motivation being to provide information to a requesting component more efficiently (pg.1, [0007], Van).

9. As to claim 2, Van discloses generating the element path identifier is implemented by a second software component communicative with the source computer program (pg.2, [0024], last 2 lines; "the hierarchical node tree is a DOM (Direct Object Model) tree").

10. As to claim 3, Van using a hierarchical tree structure representation of the graphical user interface to locate a leaf node related to the user interface element of interest (pg.2, [0024]; "node tree generator")

storing exposed identifier information of the user interface element of interest in an element path identifier data structure (pg.5, [0056], "node name", "namespace"); and

proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes of the user interface element of interest in the element path identifier data structure (pg.2, [0019]; pg.5, [0057], [0058]; Xpath or Xquery expression traverses the tree to locate the element of interest).

11. As to claim 4, Van discloses converting the element path identifier to a string type data structure (pg.4, [0056], lines 1-5; "namespace"; namespace is of string type).

12. As to claim 5, Van discloses receiving data indicative of a designated element path root node of the tree structure representation, wherein the step of proceeding up the hierarchy of the tree structure representation to store the exposed identifier



information of selected parent nodes is continued only until the element path root node is reached (pg.2, [0025]; pg.3 [0031]; the node generator generates a complete tree and store the necessary information in the XID pointer).

13. As to claim 7, Van discloses determining that at least one of the selected parent nodes is a root node of a strongly named branch portion of the tree structure representation pg.4, [0053], [0056]; namespace ensures the uniqueness of a strongly named branch portion; namespace is also used to make the tree structure more modular),

wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is guaranteed to be uniquely identifiable by a named branch element identifier all element names within a namespace must be unique); and

storing in the element path identifier data structure, the named branch element identifier for the at least one user interface element within the scope of the named branch portion (pg.5, [0056]; element that is a child of the branch portion is contained in the node name).

14. As to claim 8, Van discloses storing in the element path identifier a strong name associated with the strongly named branch along with the named branch element identifier (pg.4, [0054], last 3 lines; "hint" is stored as a part of the node name).

Art Unit: 2173

As to claim 9, the named branch element identifier stored in the element path identifier data structure is for the user interface element of interest and no identifying information related to the parent elements within the scope are stored in the element path identifier data structure (pg.5, [0056]; "namespace" encapsulates each subset of the tree into different modules; no addition information is required to distinguish local elements from different namespaces).

15. As to claims 22 and 29, Van discloses the element path identifier is a string type data structure (pg.4, [0056], lines 1-5; "namespace").

16. As to claims 23 and 30, Van discloses the element path identifier comprises exposed identifier information of component elements of an element path related to the element of interest (pg.4, [0054], last 2 lines).

17. As to claim 27, Van discloses determining that at least one node in a hierarchical tree structure representation of the graphical user interface of the source computer program is a root node of a strongly named branch portion of the tree structure representation (pg.4, [0053], [0056]; namespace ensures the uniqueness of a strongly named branch portion; namespace is also used to make the tree structure more modular).

wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is

guaranteed to be uniquely identifiable by a named branch element identifier (all element names within a namespace must be unique),

and the element path identifier comprises the named branch element identifier for the at least one user interface element within the scope of the named branch portion (pg.5, [0056]; element that is a child of the branch portion is contained in the node name).

18. As to claim 34, claim 34 incorporates substantially similar subject matter as claimed in claim 27, and is rejected under the same rationale.

**19. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Van in view of Chow and in further view of W3C XML Path Language ("Xpath", 16, Nov, 1999).**

As to claim 6, Van does not disclose the exposed identifier information is a local alpha numeric identifier not guaranteed to be unique, a class name, a module name associated with an application program or a sibling order.

In the same field of endeavor, Xpath discloses the exposed identifier information is a local alpha numeric identifier not guaranteed to be unique, a class name, a module name associated with an application program or a sibling order (section 4.1 Node Set Functions; "function: string local-name").

It would have been obvious to one of ordinary skill in the art, having the teaching of Van/Chow and Xpath before him at the time the invention was made, to modify the tree representation of GUI elements taught by Van/Chow to include XML document addressing taught by Xpath with the motivation being to provide syntax to evaluate the tree representation of GUI elements document (Xpath, section 1 Introduction.)

**20. Claims 24-26 and 31-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van in view of Chow and in further view of Alexander, III et al. (Alexander, US 6,513,155 B1).**

21. As to claims 24 and 31, Van discloses an element path identifier (pg.4, [0056], lines 1-2). Van does not disclose the element path identifier comprises class names of component elements of an element path of the element of interest.

In the same field of endeavor, Alexander discloses class names of component elements (col.10, lines 3-6).

22. It would have been obvious to one of ordinary skill in the art, having the teaching of Van/Chow and Alexander before him at the time the invention was made, to modify the tree representation of GUI elements taught by Van/Chow to include additional identification variables taught by Alexander with the motivation being to better profile the

Art Unit: 2173

element of selected element (col.2, lines 60-63).

23. As to claims 25 and 32, Van discloses an element path identifier (pg.4, [0056], lines 1-2). Van does not disclose the element path identifier comprises a module name of an application program related to component elements of an element path of the element of interest.

In the same field of endeavor, Alexander discloses module names of component elements (col.21, lines 29-33).

It would have been obvious to one of ordinary skill in the art, having the teaching of Van/Chow and Alexander before him at the time the invention was made, to modify the tree representation of GUI elements taught by Van/Chow to include additional identification variables taught by Alexander with the motivation being to better profile the element of selected element (col.2, lines 60-63).

24. As to claims 26 and 33, Van discloses an element path identifier (pg.4, [0056], lines 1-2). Van does not disclose the element path identifier comprises sibling order data.

In the same field of endeavor, Alexander discloses sibling order data (col.20, lines 22-23; "pointer to the next sibling of the node").

It would have been obvious to one of ordinary skill in the art, having the teaching of Van/Chow and Alexander before him at the time the invention was made, to modify the tree representation of GUI elements taught by Van/Chow to include additional identification variables taught by Alexander with the motivation being to better profile the element of selected element (col.2, lines 60-63).

### **Conclusion**

25. The prior art made of record on form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R. 1.111(c) to consider these references fully when responding to this action.

26. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Haoshian Shih whose telephone number is (571) 270-1257. The examiner can normally be reached on m-f 0730-1700.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2173

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

HSS

  
**TADESSE HAILU**  
*Patent Examiner*